# ESP32 Tabanlı Sıcaklık ve Nem İzleme Sistemi ile Telegram Bildirimleri

Ömer Şengül 21370031027 Bilgisayar Mühendisliği Öğrencisi

## Proje Amacı

Bu projenin amacı, bir sıcaklık ve nem izleme sistemi oluşturarak, tarımsal alanlarda (örneğin, seralar) kritik çevresel verileri izlemek ve bu verilere dayalı uyarılar göndermektir. Proje, ESP32 mikrodenetleyicisi kullanarak sıcaklık ve nem verilerini toplar ve belirli eşik değerlerinin altına düştüğünde Telegram üzerinden bildirim gönderir.

## Kullanılan Malzemeler

-ESP32 Mikrodenetleyici

-DHT11 Sıcaklık ve Nem Sensörü

-Breadboard ve jumper kablo

-Wi-Fi Bağlantısı için Kablosuz Ağ

-Telegram Bot API

## Proje Bileşenleri

## 1. ESP32 Mikrodenetleyici

ESP32, kablosuz iletişim yeteneklerine sahip bir mikrodenetleyicidir. Bu proje, Wi-Fi bağlantısı kurarak veri iletimi için kullanılmaktadır.

## 2. DHT11 Sensörü

DHT11, sıcaklık ve nem ölçümü yapabilen bir sensördür. Bu projede sıcaklık ve nem değerlerini elde etmek için kullanılmaktadır.

## 3. Telegram Bot API

Telegram, kullanıcılar arasında iletişim sağlamak için kullanılan popüler bir mesajlaşma uygulamasıdır. Proje, sıcaklık ve nem değerleri belirli eşiklerin altına düştüğünde otomatik bildirim göndermek için Telegram API'sini kullanmaktadır.

## Proje Akışı

1. Bağlantı Kurulumu: ESP32, belirlenen Wi-Fi ağına bağlanır.
2. Veri Toplama: DHT11 sensörü, sıcaklık ve nem verilerini düzenli aralıklarla toplar.
3. Veri Analizi: Toplanan veriler analiz edilerek ortalama, maksimum ve minimum değerler hesaplanır.
4. Uyarı Sistemi: Sıcaklık 10 °C'nin altına veya 35 °C'nin üstüne, nem %40'ın altına düştüğünde Telegram üzerinden bildirim gönderilir.
5. Web Arayüzü: Kullanıcılar, sıcaklık ve nem verilerini bir web arayüzü aracılığıyla gerçek zamanlı olarak izleyebilirler.

## Kod Açıklaması,

### Kütüphaneler

#include <WiFi.h>

#include <DHT.h>

#include <WebServer.h>

#include <WiFiClientSecure.h>

Bu kütüphaneler, Wi-Fi bağlantısı, sıcaklık ve nem ölçümü, web sunucusu oluşturma ve güvenli bağlantılar için kullanılmaktadır.

### Wi-Fi Bağlantısı

const char\* ssid = "Galaxy A34 5G";

const char\* password = "123123123";

WiFi.begin(ssid, password);

Burada, ESP32'nin bağlanacağı Wi-Fi ağı ve şifresi tanımlanmıştır.

### DHT Sensörünün Tanımlanması Ve Başlatılması

DHT dht(DHTPIN, DHTTYPE);

dht.begin();

DHT sensörü başlatılır ve sıcaklık/nem verileri okumaya hazır hale gelir.

### Telegram Bildirim Fonksiyonu

void sendTelegramMessage(String message) {

if (client.connect(serverName.c\_str(), 443)) {

// Mesaj gönderme işlemleri

}

}

Bu fonksiyon, belirli bir mesajı Telegram botu aracılığıyla kullanıcının chat ID'sine gönderir.

### HTML Web Arayüzü

const char\* index\_html = R"rawliteral(

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>ESP32 Sıcaklık ve Nem</title>

<style>

body { font-family: Arial, sans-serif; text-align: center; }

h1 { color: #333; }

p { font-size: 1.5em; }

</style>

</head>

<body>

<h1>ESP32 Sıcaklık ve Nem Değerleri</h1>

<p>Sıcaklık: <span id="temperature">...</span> °C</p>

<p>Nem: <span id="humidity">...</span> %</p>

<h2>Veri Analizi</h2>

<p>Ortalama Sıcaklık: <span id="avgTemperature">...</span> °C</p>

<p>Ortalama Nem: <span id="avgHumidity">...</span> %</p>

<p>Max Sıcaklık: <span id="maxTemperature">...</span> °C</p>

<p>Min Sıcaklık: <span id="minTemperature">...</span> °C</p>

<p>Max Nem: <span id="maxHumidity">...</span> %</p>

<p>Min Nem: <span id="minHumidity">...</span> %</p>

<script>

async function getData() {

const response = await fetch('/data');

const json = await response.json();

document.getElementById('temperature').innerText = json.temperature;

document.getElementById('humidity').innerText = json.humidity;

document.getElementById('avgTemperature').innerText = json.avgTemperature;

document.getElementById('avgHumidity').innerText = json.avgHumidity;

document.getElementById('maxTemperature').innerText = json.maxTemperature;

document.getElementById('minTemperature').innerText = json.minTemperature;

document.getElementById('maxHumidity').innerText = json.maxHumidity;

document.getElementById('minHumidity').innerText = json.minHumidity;

}

// Her 2 saniyede bir veriyi güncelle

setInterval(getData, 2000);

</script>

</body>

</html>

)rawliteral";

Bu kısım, kullanıcıların sıcaklık ve nem değerlerini web tarayıcısında görüntülemelerini sağlayan HTML içeriğini tanımlar. JavaScript kullanılarak, /data endpoint'inden veriler her 2 saniyede bir güncellenir.

### Veri Okuma ve JSON Formatında Gönderme

server.on("/data", []() {

temperature = dht.readTemperature();

humidity = dht.readHumidity();

// Analiz için verileri güncelle

dataCount++;

tempSum += temperature;

humSum += humidity;

if (temperature > tempMax) tempMax = temperature;

if (temperature < tempMin) tempMin = temperature;

if (humidity > humMax) humMax = humidity;

if (humidity < humMin) humMin = humidity;

// Ortalamaları hesapla

float avgTemperature = tempSum / dataCount;

float avgHumidity = humSum / dataCount;

// JSON formatında veri gönder

String json = "{";

json += "\"temperature\":" + String(temperature) + ",";

json += "\"humidity\":" + String(humidity) + ",";

json += "\"avgTemperature\":" + String(avgTemperature) + ",";

json += "\"avgHumidity\":" + String(avgHumidity) + ",";

json += "\"maxTemperature\":" + String(tempMax) + ",";

json += "\"minTemperature\":" + String(tempMin) + ",";

json += "\"maxHumidity\":" + String(humMax) + ",";

json += "\"minHumidity\":" + String(humMin);

json += "}";

server.send(200, "application/json", json);

});

Bu kod parçası, /data endpoint'i üzerinden gelen istekleri işler. DHT sensöründen sıcaklık ve nem verilerini okur, analiz eder (ortalama, maksimum, minimum değerleri hesaplar) ve bu verileri JSON formatında kullanıcıya gönderir.

### Uyarı Sistemi

if (temperature < 10) {

Serial.println("Uyarı: Sıcaklık 10 °C'nin altına düştü! Bu mahsuller için tehlikeli olabilir!");

sendTelegramMessage("Sıcaklık tehlikeli seviyelere düştü!");

}

if (temperature > 35) {

Serial.println("Uyarı: Sıcaklık 35 °C'yi geçti! Lütfen havalandırmayı açın!");

sendTelegramMessage("Sıcaklık tehlikeli seviyelere çıktı, lütfen havalandırmayı açın!");

}

if (humidity < 40) {

Serial.println("Uyarı: Nem %40'ın altında. Bu mahsuller için tehlikeli olabilir!");

sendTelegramMessage("Seranızın içindeki nem miktarı tehlikeli seviyelere düştü!");

}

Bu kısım, belirlenen eşik değerlerine göre uyarılar gönderir. Eğer sıcaklık 10 °C'nin altına veya 35 °C'nin üzerine çıkarsa ya da nem %40'ın altına düşerse, ilgili uyarı mesajları Telegram üzerinden gönderilir.

### Ana Döngü

void loop() {

// 2 saniyede bir sıcaklık ve nem verilerini seri port ekranına yazdır

if (millis() - lastTime > timerDelay) {

lastTime = millis(); // Zamanlayıcıyı güncelle

temperature = dht.readTemperature();

humidity = dht.readHumidity();

// Seri monitörde göster

Serial.print("Sıcaklık: ");

Serial.print(temperature);

Serial.print(" °C ");

Serial.print("Nem: ");

Serial.print(humidity);

Serial.println(" %");

// Uyarı sistemi

// ...

delay(5000);

}

// Web sunucusu isteğini işlemek için

server.handleClient();

}

loop fonksiyonu, sürekli olarak çalışır. Her 2 saniyede bir sıcaklık ve nem verilerini okur ve bu verileri seri monitörde görüntüler. Ayrıca, web sunucusu isteklerini işlemek için server.handleClient() çağrısını yapar.

## Sonuç

Bu proje, sıcaklık ve nem izleme sistemlerinin tarım alanında nasıl kullanılabileceğini göstermektedir. Proje, ESP32 mikrodenetleyicisi, DHT11 sensörü ve Telegram API'si kullanarak pratik bir çözüm sunmaktadır. Proje, kullanıcıların tarım alanlarındaki çevresel koşulları uzaktan izlemelerine ve kritik durumlarda hızlı bir şekilde müdahale etmelerine olanak tanımaktadır.

## Öneriler

**Sensör Kalibrasyonu**: Sensörlerin doğru ölçüm yapabilmesi için kalibrasyon işlemleri yapılmalıdır.

**Güvenlik Önlemleri**: Wi-Fi bağlantısının güvenliğini artırmak için şifreleme yöntemleri kullanılabilir.

**Veri Kaydı**: Toplanan verilerin bir veritabanında saklanması, zaman içindeki eğilimleri analiz etmek için faydalı olabilir.